

Meshless methods in dual analysis: Theoretical and implementation issues

V. Ivannikov ^{a,*}, C. Tiago ^a, J.P. Moitinho de Almeida ^a, P. Díez ^b

^a Instituto Superior Técnico, Universidade Técnica de Lisboa, Av. Rovisco Pais, 1049-001 Lisboa, Portugal

^b Departament de Matemàtica Aplicada III, Laboratori de Càlcul Numèric, Universitat Politècnica de Catalunya, Campus Norte UPC, 08034 Barcelona, Spain

ABSTRACT

This paper presents a meshless implementation of dual analysis for 2D linear elasticity problems. The derivation of the governing systems of equations for the discretized compatible and equilibrated models is detailed and crucial implementation issues of the proposed algorithm are discussed: (i) arising of deficiencies associated with the independent approximation field used for the imposition of the essential boundary conditions (EBC) for the two parts of the boundary sharing a corner and (ii) determination of the Lagrange multipliers functional space used to impose EBC. An attempt to implement the latter resulted in an approximation which is nothing more than the trace on the essential boundary of the domain nodal functions. The difficulties posed by such approximation are explained using the inf sup condition.

Several examples of global (energy) and local (displacements) quantities of interest and their bounds determination are used to demonstrate the validity of the presented meshless approach to dual analysis. Numerical assessment of the convergence rates obtained for both models is made, for different polynomial basis degrees.

1. Introduction

Meshless methods for modeling complex problems in solid mechanics are well developed. Most of them are based on strain driven approaches, where the problem's weak form is built either from equilibrium in the domain and on the static boundary or from the potential energy [4,25]. For the same problems, an alternative formulation – the so called stress driven approach – can also be used. Here the problem's weak form may be derived either by weakening the compatibility in the domain and on the kinematic boundary or from a restrained complementary potential energy statement. To the authors' knowledge, the use of the stress driven approach with meshless approximations is restricted to a single attempt [11,12].

Effectively, the implementation of this approach is more demanding than the strain driven one, mostly, due to the following characteristics: (i) higher order derivatives may appear in the arising differential operators (if, for instance, stress generating functions are used), (ii) static boundary in most practical examples is larger than the one where kinematic boundary conditions (BC) are defined and (iii) some features of the problem can easily lead to numerical

instability. These issues are not related with a specific approximation (FEM, Meshless, etc.), but are common to all of them. In fact, it results in the relative unpopularity of this approach. When both models are used in the analysis of a particular problem, it is possible to perform the so called dual analysis, introduced for the first time by Fraeijns de Veubeke with respect to the conventional finite element method [8,9]. Regardless of the high computational cost, dual analysis can be efficiently used to obtain upper and lower bounds of some characteristic values of the system, typically, total strain energy, and of local quantities in certain parts of the domain, like stresses and displacements [7].

Due to the complexity of the equilibrated model, a strategy commonly used nowadays is to recover an equilibrated solution from a compatible one. To this end various approaches have been proposed, e.g., the Ladevèze Maund element wise equilibrium recovery [15], a star patch recovery based on a partition of unity technique [19], an NS FEM technique [18], which modifies the Galerkin weak form and introduces smoothing domains inside the elements, where strains are modified in order to deliver *quasi* equilibrated solutions, etc.

Among the novelties of the paper the authors emphasize the following:

- (i) The numerical assessment of the convergence rates (for several basis degrees) of the equilibrated model for plane elasticity.

* Corresponding author. Tel.: +351 218418249.

E-mail addresses: vladimir@civil.ist.utl.pt (V. Ivannikov), carlos.tiago@ist.utl.pt (C. Tiago), moitinho@civil.ist.utl.pt (J.P. Moitinho de Almeida), pedro.diez@upc.es (P. Díez).

- (ii) The equilibrated model, together with the compatible one, allowed the authors to obtain bounds not only of the global, but also of the local quantities, using meshless approximations.
- (iii) A new proposal is made in order to locally impose the essential boundary conditions for both models. Instead of using Lagrange polynomials for the discretization of the independently approximated field on the essential boundary, this proposal is based on the trace of the domain approximation functions. This approach did not prove advantageous and its limitations were identified resorting to a set of numerical inf sup condition tests.
- (iv) A detailed analysis of the implementation issues of both models is performed.

Finally, for both models, the visualization of the stresses and of the Lagrange multipliers is presented, thus allowing a simple comparison of the solutions.

The structure of this work is as follows. Section 2 presents the basic solid mechanics differential equations and the two models required for dual analysis: compatible and equilibrated. The implementation issues, particularly relevant for the equilibrated model, are discussed in detail in Section 3. Section 4 discusses the computation of local quantities of interest by means of both models. Three numerical examples are then presented in Section 5: the classical Timoshenko beam problem, the infinite plate with a circular hole subjected to uniform tensile load and a perforated plate. Finally, Section 6 summarizes the theoretical and numerical results obtained.

2. Dual analysis

2.1. Linear elasticity equations

Consider a linear elastic plane domain, Ω , with boundary, Γ , subjected to body forces, $\bar{\mathbf{b}}$, in Ω , distributed surface loads, $\bar{\mathbf{t}}$, on the static boundary, Γ_t , and constrained by prescribed displacements, $\bar{\mathbf{u}}$, on the kinematic boundary, Γ_u . As usual, we require that $\Gamma_u \cap \Gamma_t = \emptyset$ and $\Gamma_u \cup \Gamma_t = \Gamma$.

The relations governing the plane stress elasticity problem are outlined below.

The compatibility equations in the domain are

$$\boldsymbol{\varepsilon} = \frac{1}{2}[\nabla \mathbf{u} + (\nabla \mathbf{u})^T] \quad \text{in } \Omega \quad (1)$$

and on the kinematic boundary they are

$$\mathbf{u} = \bar{\mathbf{u}} \quad \text{on } \Gamma_u, \quad (2)$$

where ∇ is the gradient operator. The equilibrium equations in the domain are

$$\text{div } \boldsymbol{\sigma} + \bar{\mathbf{b}} = \mathbf{0} \quad \text{in } \Omega \quad (3)$$

and on the static boundary they are

$$\mathbf{t} = \bar{\mathbf{t}} \quad \text{on } \Gamma_t, \quad (4)$$

where the tractions on a facet with outward normal \mathbf{n} are

$$\mathbf{t} = \boldsymbol{\sigma} \mathbf{n}. \quad (5)$$

The constitutive relation is

$$\boldsymbol{\sigma} = \mathbb{D} : \boldsymbol{\varepsilon}, \quad (6)$$

where \mathbb{D} is the fourth order constitutive tensor which, for two dimensional plane stress case, is

$$\mathbb{D} = \frac{E}{1-\nu^2} \left(\frac{1}{2} (\mathbb{I} + \mathbb{I}_T) + \nu \mathbf{I} \otimes \mathbf{I} \right). \quad (7)$$

Here several fourth order tensors [21] were introduced: the identity tensor, $(\mathbb{I})_{\alpha\beta\gamma\delta} = \delta_{\alpha\gamma}\delta_{\beta\delta}$, the transposition tensor, $(\mathbb{I}_T)_{\alpha\beta\gamma\delta} = \delta_{\alpha\delta}\delta_{\beta\gamma}$,

and the result of the dyadic product of two second order identity tensors \mathbf{I} , $(\mathbf{I} \otimes \mathbf{I})_{\alpha\beta\gamma\delta} = \delta_{\alpha\beta}\delta_{\gamma\delta}$. Notice that Greek letters, implying summation from 1 to 2, are used in the preceding equations to specify the components of the tensors.

Let us also introduce the potential energy, $\Pi = \Pi(\mathbf{u})$, and the complementary potential energy, $\Pi^* = \Pi^*(\boldsymbol{\sigma})$, both outlined below. It can be shown [10] that, under appropriate conditions, the exact value of the potential energy of the system is bounded by these two values:

$$\Pi < \Pi_{\text{exact}} < \Pi^*. \quad (8)$$

We start from the equilibrated model, making a short review of the compatible one afterwards.

2.2. Equilibrated model

To derive the associated weak form of the equilibrated model the complementary potential energy functional [10] is used:

$$\Pi^*(\boldsymbol{\sigma}) = \frac{1}{2} \int_{\Omega} \boldsymbol{\sigma} : \mathbb{D}^{-1} : \boldsymbol{\sigma} \, d\Omega - \int_{\Gamma_u} \mathbf{t} \cdot \bar{\mathbf{u}} \, d\Gamma_u, \quad (9)$$

where $\boldsymbol{\sigma}$ is an equilibrated stress field.

The static and kinematic BC are, in this case, the essential and natural ones, respectively. The latter already appear explicitly in the second term of the second member of (9), while the EBC can be imposed in various ways [13]: introduction of Lagrange multipliers [4], coupling to finite elements [5], resorting a penalty formulation [27,1], using Nitsche's method [14], etc. In the present work Lagrange multipliers are added into the complementary potential energy functional (9), resulting in the following augmented functional:

$$\bar{\Pi}^*(\boldsymbol{\sigma}, \mathbf{u}_{\Gamma_t}) = \Pi^*(\boldsymbol{\sigma}) - \int_{\Gamma_t} \mathbf{u}_{\Gamma_t} \cdot (\mathbf{t} - \bar{\mathbf{t}}) \, d\Gamma_t. \quad (10)$$

It is trivial to prove that Lagrange multipliers represent the static boundary displacements. The notation in (10) was chosen to highlight this fact.

The augmented weak form of the problem can be recovered equating the first variation of (10) to zero:

$$\int_{\Omega} \delta \boldsymbol{\sigma} : \mathbb{D}^{-1} : \boldsymbol{\sigma} \, d\Omega - \int_{\Gamma_u} (\delta \boldsymbol{\sigma} \mathbf{n}) \cdot \bar{\mathbf{u}} \, d\Gamma_u - \int_{\Gamma_t} \delta \mathbf{u}_{\Gamma_t} \cdot (\boldsymbol{\sigma} \mathbf{n} - \bar{\mathbf{t}}) \, d\Gamma_t - \int_{\Gamma_t} \mathbf{u}_{\Gamma_t} \cdot (\delta \boldsymbol{\sigma} \mathbf{n}) \, d\Gamma_t = 0. \quad (11)$$

To derive (11) we took into account (5) and the fact that variations over prescribed quantities are zero, i.e., $\delta \bar{\mathbf{u}} = \mathbf{0}$ and $\delta \bar{\mathbf{t}} = \mathbf{0}$. The weak statement (11) does not imply any special requirements for the stresses, besides the ones arise from (3).

It is possible to discretize the weak form expressed by (11) using the so called self equilibrated polynomial basis. The aim of the present work is to use meshless approximation functions. These ones cannot directly provide self equilibrated stress fields. To locally satisfy internal equilibrium (3), we consider stresses to be expressed by means of the scalar Airy stress function, Ψ [26]:

$$\boldsymbol{\sigma} = (\Delta \Psi + \bar{V}) \mathbf{I} - \nabla \otimes \nabla \Psi, \quad (12)$$

where $\Delta = \text{div}(\nabla)$ is the Laplacian and \bar{V} defines a potential of the domain body forces as follows:

$$\bar{\mathbf{b}} = -\nabla \bar{V}. \quad (13)$$

For simplicity of the forthcoming discretization we use Voigt's notation, rewriting (11) in the matrix form:

$$\int_{\Omega} \delta \boldsymbol{\sigma}^T \mathbf{D}^{-1} \boldsymbol{\sigma} \, d\Omega - \int_{\Gamma_u} (\mathbf{N} \delta \boldsymbol{\sigma})^T \bar{\mathbf{u}} \, d\Gamma_u - \int_{\Gamma_t} \delta \mathbf{u}_{\Gamma_t}^T (\mathbf{N} \boldsymbol{\sigma} - \bar{\mathbf{t}}) \, d\Gamma_t - \int_{\Gamma_t} \mathbf{u}_{\Gamma_t}^T \mathbf{N} \delta \boldsymbol{\sigma} \, d\Gamma_t = 0, \quad (14)$$

where \mathbf{N} is the boundary operator, gathering the outward normal components, i.e.,

$$\mathbf{N} = \begin{bmatrix} n_1 & 0 & n_2 \\ 0 & n_2 & n_1 \end{bmatrix} \quad (15)$$

and \mathbf{D} is the matrix form of the constitutive tensor, which for two dimensional plane stress case is

$$\mathbf{D} = \frac{E}{1-\nu^2} \begin{bmatrix} 1 & \nu & 0 \\ \nu & 1 & 0 \\ 0 & 0 & \frac{1-\nu}{2} \end{bmatrix}. \quad (16)$$

For the two dimensional case (12) can be written in a simpler manner:

$$\sigma = \begin{bmatrix} \sigma_{11} \\ \sigma_{22} \\ \sigma_{12} \end{bmatrix} = \begin{bmatrix} \Psi_{,22} \\ \Psi_{,11} \\ \Psi_{,12} \end{bmatrix} + \gamma \bar{V}, \quad \gamma = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}. \quad (17)$$

The domain and boundary variables are required to belong to the following trial and variation spaces:

$$\begin{aligned} \Psi &\in \mathbb{U}, \quad \delta\Psi \in \mathbb{U} \quad \text{where } \mathbb{U} \subset \mathcal{H}^2, \\ \mathbf{u}_{\Gamma_t} &\in \mathbb{F}, \quad \delta\mathbf{u}_{\Gamma_t} \in \mathbb{F} \quad \text{where } \mathbb{F} \subset \mathcal{H}^{-1/2}, \end{aligned} \quad (18)$$

\mathcal{H}^2 and $\mathcal{H}^{-1/2}$ being the corresponding Sobolev spaces.

Let us introduce a discretization for the Airy stress function and its variation in matrix notation:

$$\Psi^h = \Phi \tilde{\Psi} \quad \text{and} \quad \delta\Psi^h = \Phi \delta\tilde{\Psi}, \quad (19)$$

where

$$\Phi = [\phi_1 \ \phi_2 \ \dots \ \phi_n] \quad (20)$$

is an n row vector of approximation functions. Substituting the latter approximation for Ψ in (17) and taking into account that $\delta\bar{V} = 0$ we obtain the discrete forms of the domain stresses and their variations for the two dimensional case:

$$\begin{aligned} \sigma^h &= \mathbf{C} \tilde{\Psi} + \gamma \bar{V} \\ \delta\sigma^h &= \mathbf{C} \delta\tilde{\Psi} \end{aligned} \quad \text{where } \mathbf{C} = \begin{bmatrix} \Phi_{,22} \\ \Phi_{,11} \\ \Phi_{,12} \end{bmatrix} \quad (21)$$

and $\tilde{\Psi}$ is a vector that gathers the weights of the approximation. We also introduce an approximation for the Lagrange multipliers and their variations:

$$\mathbf{u}_{\Gamma_t}^h = \mathbf{L} \tilde{\mathbf{u}}_{\Gamma_t} \quad \text{and} \quad \delta\mathbf{u}_{\Gamma_t}^h = \mathbf{L} \delta\tilde{\mathbf{u}}_{\Gamma_t}. \quad (22)$$

Discretized approximation spaces for the domain and boundary parameters are

$$\begin{aligned} \Psi^h &\in \mathbb{U}_h, \quad \delta\Psi^h \in \mathbb{U}_h \quad \text{where } \mathbb{U}_h \subset \mathbb{U}, \\ \mathbf{u}_{\Gamma_t}^h &\in \mathbb{F}_h, \quad \delta\mathbf{u}_{\Gamma_t}^h \in \mathbb{F}_h \quad \text{where } \mathbb{F}_h \subset \mathbb{F}. \end{aligned} \quad (23)$$

Substituting (21) and (22) into (11) and after some simplifications, it is possible to obtain

$$\begin{aligned} &\delta\tilde{\Psi}^T \int_{\Omega} \Phi^T \mathbf{D}^{-1} \Phi \, d\Omega \, \tilde{\Psi} + \delta\tilde{\Psi}^T \int_{\Omega} \Phi^T \mathbf{D}^{-1} \gamma \bar{V} \, d\Omega \\ &\quad \delta\tilde{\Psi}^T \int_{\Gamma_u} \Phi^T \mathbf{N}^T \bar{\mathbf{u}} \, d\Gamma_u - \delta\tilde{\mathbf{u}}_{\Gamma_t}^T \int_{\Gamma_t} \mathbf{L}^T \mathbf{N} \Phi \, d\Gamma_t \, \tilde{\Psi} \\ &\quad + \delta\tilde{\mathbf{u}}_{\Gamma_t}^T \int_{\Gamma_t} \mathbf{L}^T \bar{\mathbf{t}} \, d\Gamma_t - \delta\tilde{\Psi}^T \int_{\Gamma_t} \Phi^T \mathbf{N}^T \mathbf{L} \, d\Gamma_t \, \tilde{\mathbf{u}}_{\Gamma_t} \\ &\quad \delta\tilde{\mathbf{u}}_{\Gamma_t}^T \int_{\Gamma_t} \mathbf{L}^T \mathbf{N} \gamma \bar{V} \, d\Gamma_t = 0. \end{aligned} \quad (24)$$

Since the weak form (24) must hold for $\forall \delta\tilde{\Psi}$ and $\forall \delta\tilde{\mathbf{u}}_{\Gamma_t}$, we can finally write the governing system of equations:

$$\begin{bmatrix} \mathbf{F} & \mathbf{G} \\ \mathbf{G}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\Psi} \\ \tilde{\mathbf{u}}_{\Gamma_t} \end{bmatrix} = \begin{bmatrix} \mathbf{d} \\ \mathbf{f} \end{bmatrix}, \quad (25)$$

where

$$\mathbf{F} = \int_{\Omega} \Phi^T \mathbf{D}^{-1} \Phi \, d\Omega, \quad (26a)$$

$$\mathbf{G} = \int_{\Gamma_t} \Phi^T \mathbf{N}^T \mathbf{L} \, d\Gamma_t, \quad (26b)$$

$$\mathbf{d} = \int_{\Gamma_u} \Phi^T \mathbf{N}^T \bar{\mathbf{u}} \, d\Gamma_u - \int_{\Omega} \Phi^T \mathbf{D}^{-1} \gamma \bar{V} \, d\Omega, \quad (26c)$$

$$\mathbf{f} = \int_{\Gamma_t} \mathbf{L}^T \mathbf{N} \gamma \bar{V} \, d\Gamma_t - \int_{\Gamma_t} \mathbf{L}^T \bar{\mathbf{t}} \, d\Gamma_t. \quad (26d)$$

2.3. Compatible model

Following an approach that is complementary to the previous model, the compatibility relations are satisfied *a priori*, instead of the equilibrium ones. To derive the required weak form the potential energy functional is used:

$$\Pi(\mathbf{u}) = \frac{1}{2} \int_{\Omega} \boldsymbol{\varepsilon} : \mathbb{D} : \boldsymbol{\varepsilon} \, d\Omega - \int_{\Gamma_t} \mathbf{u} \cdot \bar{\mathbf{t}} \, d\Gamma_t, \quad (27)$$

where \mathbf{u} is a compatible displacement field.

Complementary to the equilibrated model, the kinematic and static boundary conditions are the essential and natural ones, respectively. The latter are explicitly added as a second term in (27) and the essential boundary conditions can be imposed in various ways [13]. In the present work Lagrange multipliers are added into the potential energy (27), resulting in the following augmented potential energy functional:

$$\bar{\Pi}(\mathbf{u}, \mathbf{t}_{\Gamma_u}) = \Pi(\mathbf{u}) - \int_{\Gamma_u} \mathbf{t}_{\Gamma_u} \cdot (\mathbf{u} - \bar{\mathbf{u}}) \, d\Gamma_u. \quad (28)$$

From the physics of the problem it follows that Lagrange multipliers represent the kinematic boundary tractions. The notation in (28) was chosen to reflect this fact.

The augmented weak form of the problem can be recovered by equating variation of (28) to zero [24]:

$$\begin{aligned} &\int_{\Omega} \delta\boldsymbol{\varepsilon} : \mathbb{D} : \boldsymbol{\varepsilon} \, d\Omega - \int_{\Omega} \bar{\mathbf{b}} \cdot \delta\mathbf{u} \, d\Omega - \int_{\Gamma_t} \bar{\mathbf{t}} \cdot \delta\mathbf{u} \, d\Gamma_t \\ &\quad \int_{\Gamma_u} \delta\mathbf{t}_{\Gamma_u} \cdot (\mathbf{u} - \bar{\mathbf{u}}) \, d\Gamma_u - \int_{\Gamma_u} \mathbf{t}_{\Gamma_u} \cdot \delta\mathbf{u} \, d\Gamma_u = 0. \end{aligned} \quad (29)$$

The approximants for the domain displacements and unknown boundary tractions are defined in the similar manner as it was done above for the equilibrated model, being the only difference now the fact that the domain approximation (for the displacements) has two degrees of freedom instead of one. Variables of both types belong to the following trial and variation spaces:

$$\begin{aligned} \mathbf{u} &\in \mathbb{V}, \quad \delta\mathbf{u} \in \mathbb{V} \quad \text{where } \mathbb{V} \subset \mathcal{H}^1, \\ \mathbf{t}_{\Gamma_u} &\in \mathbb{F}, \quad \delta\mathbf{t}_{\Gamma_u} \in \mathbb{F} \quad \text{where } \mathbb{F} \subset \mathcal{H}^{-1/2}. \end{aligned} \quad (30)$$

Switching again to Voigt's notation, the displacements \mathbf{u}^h and their variations $\delta\mathbf{u}^h$ are approximated as

$$\mathbf{u}^h = \Phi \tilde{\mathbf{u}} \quad \text{and} \quad \delta\mathbf{u}^h = \Phi \delta\tilde{\mathbf{u}}, \quad (31)$$

where

$$\Phi = \begin{bmatrix} \phi_1 & 0 & \phi_2 & 0 & \dots & \phi_n & 0 \\ 0 & \phi_1 & 0 & \phi_2 & \dots & 0 & \phi_n \end{bmatrix} \quad (32)$$

is a $2 \times 2n$ matrix of approximation functions. The essential boundary tractions, \mathbf{t}^h , have the approximation gathered in matrix \mathbf{L} :

$$\mathbf{t}_{\Gamma_u}^h = \mathbf{L}\tilde{\mathbf{t}}_{\Gamma_u} \quad \text{and} \quad \delta \mathbf{t}_{\Gamma_u}^h = \mathbf{L}\delta \tilde{\mathbf{t}}_{\Gamma_u}. \quad (33)$$

Discretized domain, \mathbf{u}^h , and boundary, $\mathbf{t}_{\Gamma_u}^h$, approximations should belong to the following spaces:

$$\begin{aligned} \mathbf{u}^h &\in \mathbb{V}_h, & \delta \mathbf{u}^h &\in \mathbb{V}_h & \text{where } \mathbb{V}_h &\subset \mathbb{V}, \\ \mathbf{t}_{\Gamma_u}^h &\in \mathbb{F}_h, & \delta \mathbf{t}_{\Gamma_u}^h &\in \mathbb{F}_h & \text{where } \mathbb{F}_h &\subset \mathbb{F}. \end{aligned} \quad (34)$$

Eq. (29) leads to the following system:

$$\begin{bmatrix} \mathbf{K} & \mathbf{G} \\ \mathbf{G}^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{u}} \\ \tilde{\mathbf{t}}_{\Gamma_u} \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ \mathbf{v} \end{bmatrix}, \quad (35)$$

being

$$\mathbf{K} = \int_{\Omega} \mathbf{B}^T \mathbf{D} \mathbf{B} \, d\Omega, \quad (36a)$$

$$\mathbf{G} = \int_{\Gamma_u} \Phi^T \mathbf{L} \, d\Gamma_u, \quad (36b)$$

$$\mathbf{f} = \int_{\Omega} \Phi^T \bar{\mathbf{b}} \, d\Omega + \int_{\Gamma_t} \Phi^T \bar{\mathbf{t}} \, d\Gamma_t, \quad (36c)$$

$$\mathbf{v} = \int_{\Gamma_u} \mathbf{L}^T \bar{\mathbf{u}} \, d\Gamma_u, \quad (36d)$$

where

$$\mathbf{B} = \partial \Phi. \quad (37)$$

Here ∂ is the compatibility operator, written for the 2D case as

$$\partial = \begin{bmatrix} \frac{\partial}{\partial x_1} & 0 \\ 0 & \frac{\partial}{\partial x_2} \\ \frac{\partial}{\partial x_1} & \frac{\partial}{\partial x_2} \end{bmatrix}. \quad (38)$$

3. Implementation issues

The meshless methods implementation issues, crucial for the equilibrated model and the subsequent dual analysis, are discussed in the current section. General implementation aspects of the compatible one can be found in [22], where a complete and profound review is given.

3.1. Domain approximation functions

Let \mathbf{x}_i be the coordinates of a particle i in a domain Ω containing n nodes. Then the Multiple Fixed Least Squares (MFLS) approximation $f^h(\mathbf{x})$ for an arbitrary function $\bar{f}(\mathbf{x})$ is [20,4]

$$f^h(\mathbf{x}) = \Phi(\mathbf{x})\bar{\mathbf{f}}, \quad (39)$$

where

$$\Phi(\mathbf{x}) = \mathbf{p}^T(\mathbf{x})\mathbf{A}(\mathbf{x})^{-1}\mathbf{B}(\mathbf{x}), \quad (40)$$

being

$$\mathbf{A}(\mathbf{x}) = \mathbf{P}^T \mathbf{V}(\mathbf{x}) \mathbf{P} \quad \text{and} \quad \mathbf{B}(\mathbf{x}) = \mathbf{P}^T \mathbf{V}(\mathbf{x}). \quad (41)$$

Matrix \mathbf{P} contains values of basis functions $\mathbf{p}(\mathbf{x})$ for each particle of a domain:

$$\mathbf{P} = [\mathbf{p}(\mathbf{x}_1) \quad \mathbf{p}(\mathbf{x}_2) \quad \dots \quad \mathbf{p}(\mathbf{x}_n)]^T. \quad (42)$$

The basis functions are defined as polynomials \mathbf{x}^k of degree k , but any set of linearly independent functions can be used.

In (41) we also introduced

$$\mathbf{V} = \text{diag}[w(r_1), w(r_2), \dots, w(r_n)], \quad (43)$$

where

$$r_i = \frac{\|\mathbf{x}_i - \mathbf{x}\|}{h_i} \quad (44)$$

and h_i is the support radius of the i th particle. To specify \mathbf{V} a weight function, $w(r)$, needs to be defined. For this particular case we chose a quintic spline function with circular support:

$$w(r) = \begin{cases} 1 & 10r^2 + 20r^3 & 15r^4 + 4r^5 & \text{if } r \leq 1 \\ 0 & & & \text{if } r > 1. \end{cases} \quad (45)$$

Though the Multiple Fixed Least Squares approximation [23] is used, it can be shown that all the conclusions obtained also hold for the MLS case.

3.2. Computation of the support radius

According to the properties of approximation functions (39), the support size influences significantly the sparsity of the stiffness and flexibility matrices and, consequently, the computational cost. Hence, from the practical point of view, it is desirable to choose the smallest possible support. However, its minimal size is bounded by the requirement of generating a non-singular moment matrix, \mathbf{A} , defined in (41).

Consider the usage of polynomial basis of degree $k > 0$. When dealing with 2D problems, in order to avoid matrix ill conditioning, supports must satisfy the following conditions [24,22]:

- (i) each point of interest must be covered, at least, by $\frac{1}{2}(k+1)$ ($k+2$) supports from its neighborhood;
- (ii) a certain point of interest must be covered in each two non-collinear directions by, at least, m supports.

When these requirements are satisfied, it is possible, for a given discretization, to adjust automatically the sizes of the supports in order to guaranty matrix \mathbf{A} invertibility and provide maximum sparsity. During this procedure each support size is enlarged according to a “safety” factor d , which normally ranges from 1.1 to 2.5 and in most examples below is considered to be $d=1.5$.

For the compatible model to reproduce constant stress field we have to use, at least, a linear basis for the approximation of the domain displacements, since stresses are their first derivatives. To reproduce such field with the equilibrated solution the basis must be, at least, quadratic which results in a loss of sparsity. For instance, to obtain an exact solution of the Timoshenko beam problem, exposed in Section 5.1, it is required to use a 3rd degree basis for the compatible model and a 4th degree for the equilibrated one.

3.3. Lagrange multipliers approximation functions

To approximate the Lagrange multipliers two types of approximants were used within the proposed approach:

- (i) Lagrange polynomials of a prescribed degree:

$$L(s) = \sum_{j=0}^q L_j(s) y_j \quad (46)$$

where

$$L_j(s) = \prod_{\substack{0 \leq p \leq q \\ p \neq j}} \frac{s - s_p}{s_j - s_p} \quad (47)$$

- and s_p , $p=1,2,\dots,q$, are the coordinates of the boundary nodes, used for discretization;
- (ii) the trace of the domain approximation functions on the essential boundary, Γ_e , i.e.

$$\mathbf{L} = \Phi|_{\Gamma_e}. \quad (48)$$

The latter idea was originally proposed in [13], but was not implemented and numerically assessed in that work.

These two approaches have completely different fundamental behaviors. The first one implies that different spaces to approximate values within the domain and on the essential boundary are used. Provided these spaces are chosen properly, this approach will lead to a numerically stable scheme. However, the drawback of such choice is that boundary conditions are not imposed locally, i.e., are not satisfied exactly at each point of the essential boundary.

On the other hand, it can be shown [24] that the exact imposition of the essential boundary conditions is guaranteed when: (i) both spaces for the approximation function inside a domain and on a boundary coincide and (ii) the domain approximation functions are able to exactly reproduce the imposed displacements. In practice MFLS approximants cannot be used explicitly in this manner. For instance, consider the domain in Fig. 1a clamped on the left side and subjected to a unit tensile load, uniformly distributed on the right side. Material parameters are Young's modulus $E = 1 \times 10^3$ and Poisson's ratio $\nu = 0.3$.

This problem was analyzed with the compatible model and the two presented essential boundary approximants. A first degree basis was used for the domain approximation. The results presented in Fig. 2a were obtained with first degree Lagrange polynomials, i.e., using $q=1$ in (46), for boundary approximation, as defined in Section 3.6.

When approximation (48) is used, the deformed shape depicted in Fig. 2b is obtained, where it can be seen that zero value displacements propagate inside a domain, clamping not only the boundary displacements along the line $x_1=0$, but also the vertical displacements of the area where the domain unknowns

are approximated by the particles whose supports intersect the essential boundary. It is visible that exactly two lines of particles intersect the boundary and the area bounded by the second line is completely locked, see Fig. 1b, where interacting particles are marked with red stars, while the green color is used for noninteracting ones. Notice that only 3 particles, out of the 12 nearest to the boundary, are shown for comparison in order to make the figure simpler.

To explain this result let us write the discrete bilinear form related to systems (35) and (25) [2]:

$$\begin{cases} a(\mathbf{u}_h, \mathbf{v}_h) + b(\mathbf{v}_h, \phi_h) = f(\mathbf{v}_h) & \forall \mathbf{v}_h \in \mathbb{V}_h \\ b(\mathbf{u}_h, \psi_h) = g(\psi_h) & \forall \psi_h \in \mathbb{F}_h, \end{cases} \quad (49)$$

where \mathbf{u}_h and ϕ_h form an arbitrary solution of (49) and \mathbf{v}_h and ψ_h are corresponding trial functions. For this case, the LBB condition reads [3]

$$\inf_{\psi_h \in \mathbb{F}_h} \sup_{\mathbf{v}_h \in \mathbb{V}_h} \frac{b(\mathbf{v}_h, \psi_h)}{\|\mathbf{v}_h\|_{\Omega} \|\psi_h\|_{\Gamma}} = \beta_h \geq \beta > 0, \quad (50)$$

where β is a positive constant independent of h , the discretization size parameter. $\mathbb{V}_h \subset \mathcal{H}^1$ is the domain variable approximation space, the first order Hilbert space, and \mathbb{F}_h is the boundary variables approximation space. The denominator of (50) contains norms which can be introduced in a different manner. To simplify further derivations we will use L_2 norms for both approximation spaces:

$$\|\mathbf{v}\|_0^2 = (\mathbf{v}_h, \mathbf{v}_h)_{L_2(\Omega)} = \int_{\Omega} \sum_i^2 (v_i)^2 d\Omega, \quad (51a)$$

$$\|\psi\|_0^2 = (\psi_h, \psi_h)_{L_2(\Gamma)} = \int_{\Gamma} \sum_i^2 (\psi_i)^2 d\Gamma. \quad (51b)$$

The last ingredient of the $\inf \sup$ condition, the numerator of (50), can be expressed as

$$b(\mathbf{v}_h, \psi_h) = (\mathbf{J}\mathbf{v}_h, \psi_h)_{L_2(\Gamma)} = \int_{\Gamma} \sum_i^2 (v_i \psi_i) d\Gamma, \quad (52)$$

where \mathbf{J} is an operator which projects domain approximation space on the boundary.

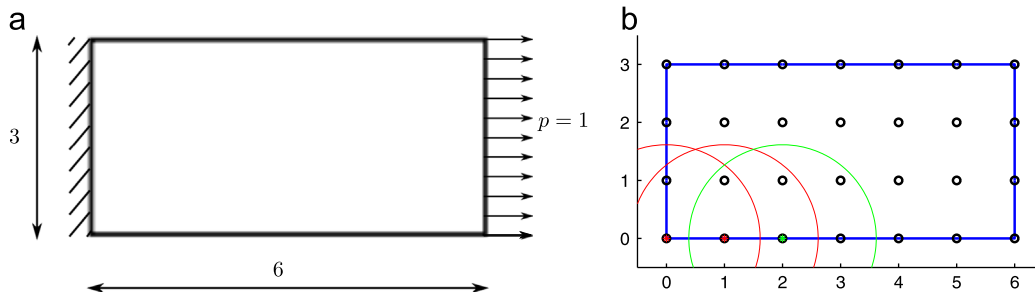


Fig. 1. Tension test. (a) Problem statement. (b) Particles distribution. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

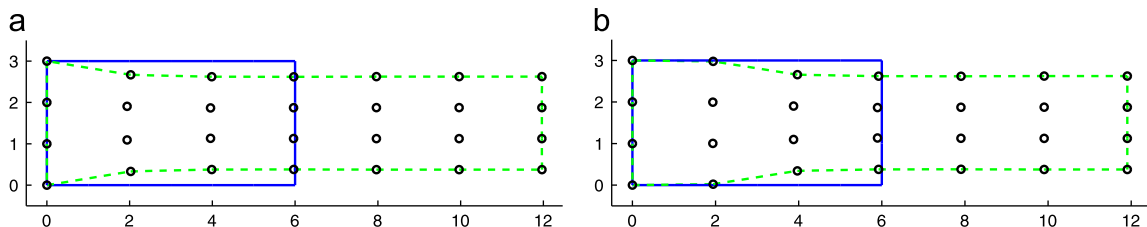


Fig. 2. Comparison of Lagrange multipliers approximation functions. (a) Lagrange polynomials. (b) MFLS approximation functions.

The entire analytical proof of the inf sup condition is extremely difficult due to the complex nature of the interpolation functions used in the current implementation of the meshless method. So, for our scheme we proceed with a numerical inf sup test [2], where we compute the inf sup parameter β_h for a sequence of discretizations of a certain problem. The problem is stable once the inf sup parameter asymptotically approaches a positive value greater than zero as the discretization is refined.

According to [2,6] the inf sup condition can be rewritten in the matrix notation. To this end we rewrite the components of (50) in the corresponding form:

$$\|\mathbf{v}_h\|_0^2 = \mathbf{V}_h^T \mathbf{M}_h^Q \mathbf{V}_h, \quad (53a)$$

$$\|\boldsymbol{\psi}_h\|_0^2 = \boldsymbol{\Psi}_h^T \mathbf{M}_h^F \boldsymbol{\Psi}_h, \quad (53b)$$

$$b(\mathbf{v}_h, \boldsymbol{\psi}_h) = \mathbf{V}_h^T \mathbf{G}_h \boldsymbol{\Psi}_h, \quad (53c)$$

where \mathbf{V}_h and $\boldsymbol{\Psi}_h$ are vectors of nodal values corresponding to \mathbf{v}_h and $\boldsymbol{\psi}_h$ respectively, and \mathbf{M}_h^Q , \mathbf{M}_h^F and \mathbf{G}_h are matrices, introduced to compute the norms of the corresponding discrete values:

$$\mathbf{M}_h^Q = \int_{\Omega} \boldsymbol{\Phi}^T \boldsymbol{\Phi} d\Omega, \quad (54a)$$

$$\mathbf{M}_h^F = \int_{\Gamma} \mathbf{L}^T \mathbf{L} d\Gamma, \quad (54b)$$

$$\mathbf{G}_h = \int_{\Gamma} \boldsymbol{\Phi}^T \mathbf{L} d\Gamma. \quad (54c)$$

Notice that (54a) and (54b) are mass matrices (this fact has been reflected in the chosen notation) and the matrix (54c) coincides, apart from the sign, with matrix \mathbf{G} introduced in (36).

Assume that there exists \mathbf{w}_h , such that $\boldsymbol{\psi}_h = \mathbf{J}\mathbf{w}_h$, where \mathbf{w}_h belongs to the domain approximation space, as well as \mathbf{v}_h . In this case it can be shown [2] that (52) may be written as

$$b(\mathbf{v}_h, \boldsymbol{\psi}_h) = (\mathbf{J}\mathbf{v}_h, \mathbf{J}\mathbf{w}_h)_{L_2(\Gamma)} \quad (55)$$

and its matrix form (53c) as

$$b(\mathbf{v}_h, \boldsymbol{\psi}_h) = \mathbf{V}_h^T \mathbf{R}_h \mathbf{W}_h, \quad (56)$$

being

$$\mathbf{R} = \mathbf{G}_h \mathbf{M}_h^F^{-1} \mathbf{G}_h^T, \quad (57)$$

where vector \mathbf{W}_h , similar to \mathbf{V}_h , is a vector of nodal values corresponding to $\mathbf{w}_h \in \mathbb{V}_h$. Consequently (53b) will take the form

$$b(\mathbf{v}_h, \boldsymbol{\psi}_h) = \mathbf{V}_h^T \mathbf{R}_h \mathbf{W}_h^T. \quad (58)$$

With the aid of these simplifications the inf sup condition (50) may be written in the following matrix form:

$$\inf_{\mathbf{W}_h} \sup_{\mathbf{V}_h} \frac{\mathbf{V}_h^T \mathbf{R}_h \mathbf{W}_h}{\sqrt{\mathbf{V}_h^T \mathbf{M}_h^Q \mathbf{V}_h} \sqrt{\mathbf{W}_h^T \mathbf{R}_h \mathbf{W}_h}} = \beta_h \geq \beta > 0. \quad (59)$$

According to [2, p. 324], the inf sup parameter β_h is given by the square root of the smallest nonzero eigenvalue of the problem

$$\mathbf{R}_h \mathbf{u}_h = \lambda \mathbf{M}_h^Q \mathbf{u}_h. \quad (60)$$

Hence, if there are $(k-1)$ zero eigenvalues (since \mathbf{M}_h^Q is a positive semidefinite matrix) and we sort the eigenvalues in ascending order, we find that

$$\beta_h = \sqrt{\lambda_k}. \quad (61)$$

In practice, problem (60) has a very large size and, as has just been mentioned, lots of zero eigenvalues. To eliminate those eigenvalues and reduce the size of the problem the equivalent problem may be posed [6]:

$$\mathbf{R}_h' \mathbf{q}_h = \lambda \mathbf{M}_h^F \mathbf{q}_h, \quad (62)$$

where

$$\mathbf{R}_h' = \mathbf{G}_h^T \mathbf{M}_h^Q^{-1} \mathbf{G}_h. \quad (63)$$

This problem has no zero eigenvalues and the smallest one will give the value of the inf sup parameter.

Let us recall the problem proposed in Fig. 1a and solve it with four nested distributions of particles. A linear basis was chosen in order to involve a minimum number of domain particles in the boundary approximation (see discussion in Section 3.2). In order to perform numerical test with representative results, special attention should be given to the generation of the domain particle distribution. For instance, if domain is discretized using a regular arrangement of nodes, like the one from Fig. 3a, the smallest eigenvalue of the problem is extremely low, 6.45×10^{-6} , and once the discretization is refined, this value does not decrease, but oscillates around its initial number. Such behavior arises since the magnitude of the eigenvalue is small, being easily affected by numerical errors, which come out mainly from the numerical integration procedure and the eigenvalue problem solver. In fact, appearance of eigenvalues of this order of magnitude, 10^{-6} – 10^{-7} , already allows us to conclude that the problem is not stable, but to be fully consistent with the idea of the numerical inf sup test it is desirable to observe the smallest eigenvalue decreasing, rather than being so low from the first discretization.

The magnitude of smallest eigenvalue for the discretization (Fig. 3a) is specified by the nature of the approximation functions built by the proposed meshless method. Since they decrease suddenly when approaching the border of their supports, the trace of the second line of the particles has a very small value (see traces 4, 5 and 6 in Fig. 4a): the maximum value is of the order of magnitude of 10^{-3} , which square gives something $\sim 10^{-6}$, what has been observed in the numerical test. Moreover, if one more line of particles intersects the boundary, its trace will be even smaller. Notice that the requirements from Section 3.2 imply that the boundary is always intersected by at least two lines of particles.

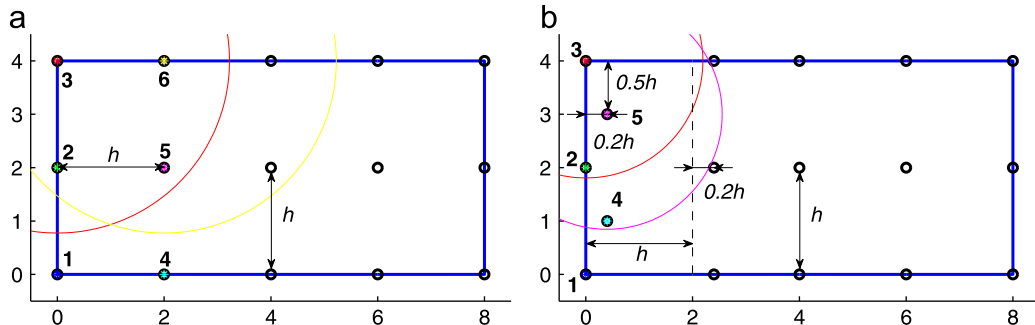


Fig. 3. (a) Regular and (b) modified regular discretizations for the inf-sup test.

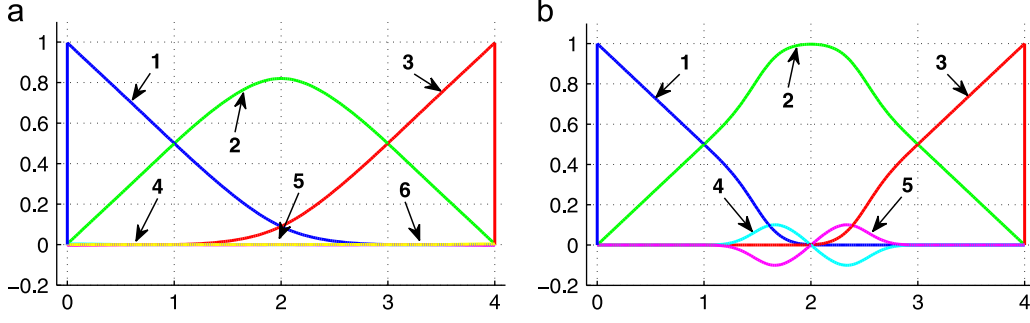


Fig. 4. Boundary traces for (a) regular and (b) modified regular discretizations.

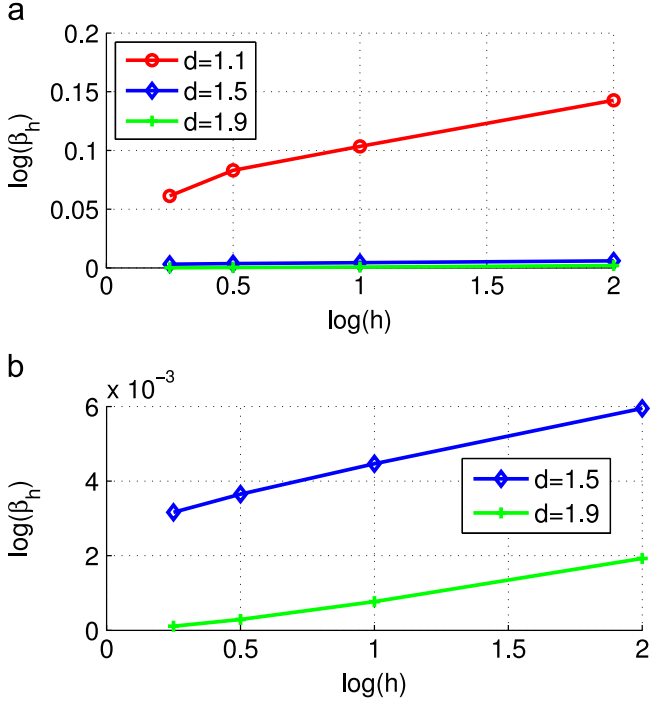


Fig. 5. Inf-sup test results for various supports sizes. (a) Inf-sup test for $d = 1.1$, $d = 1.5$, $d = 1.9$. (b) Inf-sup test for $d = 1.5$, $d = 1.9$ (enlarged).

To overcome this difficulty, another type of discretization is proposed (see Fig. 3b), where an extra line of particles was placed near the boundary. This discretization gives the trace of the second line of particles of the order of magnitude of 10^{-1} (see traces 4 and 5 in Fig. 4b), which makes the smallest eigenvalue less sensitive to numerical errors. Positions of the extra particles depend on the average distance between particles, h . Therefore, a sequence of meshes of this kind is more suitable to track the behavior of the smallest eigenvalue.

The result of the inf-sup test depends only on the topology of the domain and on the chosen approximations. It is not related to the model being considered. The test was performed for basis degree $k=1$ with various support sizes, which were adjusted by means of a factor $d=1.1 \dots 1.9$ (see Section 3.2). As expected, if large supports are considered, more domain particles are involved in the boundary approximation: 5 for $d=1.1$, 8 for $d=1.5$, 11 for $d=1.9$, and the most distant ones lead to traces with very low maximum values. This results in lower values of the inf-sup parameter $\log(\beta_h)$, however, in Fig. 5 a tendency of its steady decrease is

still observed for all cases, meaning that the analyzed discretization schemes are said to fail the inf-sup test. The results confirm that the choice of traces of meshless approximation functions for the boundary values approximants for the problems presented cannot lead to a stable solution scheme.

3.4. Rank deficiency of the flexibility matrix

Both the flexibility matrix (26a) and the stiffness matrix (36a) are rank deficient, since they do not include any information regarding the essential boundary conditions. In the compatible model they are imposed directly on the displacements which are the domain unknowns, but for the equilibrated model they are imposed on the stresses, which do not coincide with the domain parameters (the weights of the Airy stress function approximation). Hence, for the stress function three rigid body modes are not restrained, even after the essential boundary conditions have been imposed. To eliminate this ill conditioning, extra conditions have to be imposed on the Airy function, which can be set to arbitrary values at any three noncollinear points. Effectively, if $\Psi(x_1, x_2)$ is a solution of (25), then $\underline{\Psi}(x_1, x_2) = \Psi(x_1, x_2) + ax_1 + bx_2 + c$ is also a solution of the system, where a , b and c are arbitrary constants, since

$$\frac{\partial^2 \underline{\Psi}(x_1, x_2)}{\partial x_1^2} = \frac{\partial^2 \Psi(x_1, x_2)}{\partial x_1^2}, \quad (64a)$$

$$\frac{\partial^2 \underline{\Psi}(x_1, x_2)}{\partial x_1 \partial x_2} = \frac{\partial^2 \Psi(x_1, x_2)}{\partial x_1 \partial x_2}, \quad (64b)$$

$$\frac{\partial^2 \underline{\Psi}(x_1, x_2)}{\partial x_2^2} = \frac{\partial^2 \Psi(x_1, x_2)}{\partial x_2^2}. \quad (64c)$$

These restrictions are imposed by means of discrete Lagrange multipliers at the aforementioned points:

$$\lambda_i^{\underline{\Psi}}(\Psi_i - \bar{\Psi}_i) = 0, \quad (65)$$

where $\bar{\Psi}_i$ is the imposed value of the Airy stress function on the i th point, $i = \{1, 2, 3\}$.

3.5. Linear dependencies on the essential boundary

When imposing essential boundary conditions, one should take care of possible linear dependencies which arise in vertices connecting two adjacent boundaries. This problem appears in both models. We present it for the compatible one only.

In Fig. 6 two boundaries with prescribed boundary conditions are connected at point A. Since this point belongs to discretization of both boundaries its displacements are constrained twice and Lagrange multiplier vectors $\lambda_A^{f_{i_1}}$ and $\lambda_A^{f_{i_2}}$ turn out to be linearly

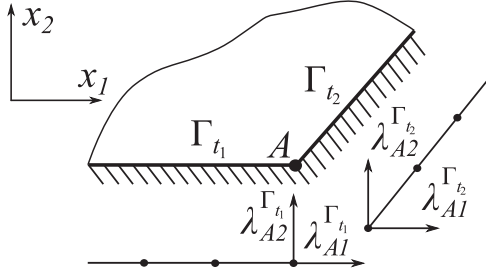


Fig. 6. Dependencies in Lagrange multipliers on a boundary.

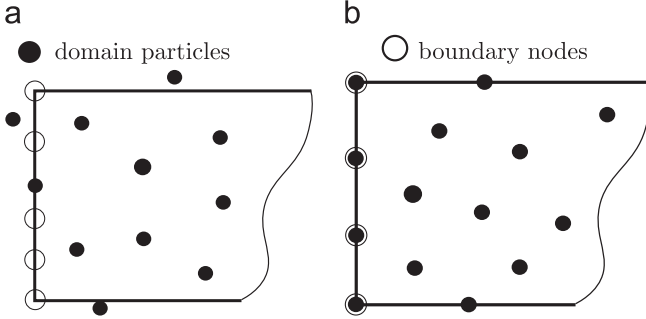


Fig. 7. Boundary and domain approximations. (a) Independent domain and boundary approximations. (b) Consistent domain and boundary approximations.

dependent. To release this constraint we add extra Lagrange multipliers in the following manner:

$$\alpha(\lambda_A^{\Gamma_{t_1}} \lambda_A^{\Gamma_{t_2}}) = 0. \quad (66)$$

We also note that it is necessary to ensure the admissibility of nonhomogeneous boundary conditions. For example, at point A, the horizontal displacement in Γ_{t_1} must coincide with the one in Γ_{t_2} . The complementary conditions, which constrain imposed tractions, apply to the equilibrated model.

3.6. Relation between domain and boundary approximations

The approximation of the Lagrange multipliers should be carefully chosen in order to provide an optimal number of boundary degrees of freedom. On one hand, if the number of boundary constraints is too low, the EBC are imposed insufficiently and, on the other hand, if the boundary is overconstrained, it leads to the ill conditioning of the governing system of equations. In general, domain and boundary approximations are totally independent, so the latter one can be built separately (see Fig. 7a). Our experience shows that one of the most optimal and reliable ways to build the boundary approximation is to make it coincident with the domain one, i.e., those domain particles, which belong to the essential boundary, are used as boundary nodes for Lagrange multipliers approximation (see Fig. 7b). This idea is followed throughout all examples.

4. Computation of local quantities

Besides the computation of bounds of the strain energy, dual analysis can also be applied in the evaluation of upper and lower bounds for local quantities of interest. According to [7], a local

quantity is a weighted integral of a field in a sub region. Most common choice for 2D problems is to consider computing of local quantities along a certain part of the boundary. The weighting function can be considered as a virtual action, therefore the local quantity can be computed using the principle of virtual work, resulting in

$$\begin{aligned} \sigma_{local} &= \int_{\hat{\Gamma}_u} \hat{\mathbf{u}}_r \cdot \mathbf{t} \, d\Gamma \\ &= \int_{\Omega} \hat{\boldsymbol{\varepsilon}}_c : \boldsymbol{\sigma} \, d\Omega + \int_{\Omega} \hat{\mathbf{u}}_c \cdot \mathbf{b} \, d\Omega + \int_{\Gamma_t} \hat{\mathbf{u}}_c \cdot \mathbf{t}_r \, d\Gamma \end{aligned} \quad (67)$$

and

$$\delta_{local} = \int_{\hat{\Gamma}_t} \mathbf{u} \cdot \hat{\mathbf{t}}_r \, d\Gamma = \int_{\Omega} \boldsymbol{\varepsilon} : \hat{\boldsymbol{\sigma}}_e \, d\Omega - \int_{\Gamma_u} \mathbf{u}_r \cdot \hat{\mathbf{t}}_e \, d\Gamma, \quad (68)$$

where a hat is used for the virtual actions and solutions, subscript “e” refers to equilibrated and “c” refers to compatible values.

We only demonstrate the computation of displacements for homogeneous kinematic boundary conditions. For this particular case (68) can be simplified by eliminating the second term. We also introduce the following integrals, which represent the same quantity computed by means of combining solutions of different types:

$$\delta^{\hat{e}e} = \int_{\Omega} \hat{\boldsymbol{\varepsilon}}_e : \boldsymbol{\sigma}_e \, d\Omega \quad (69)$$

and

$$\delta^{\hat{e}c} = \int_{\Omega} \hat{\boldsymbol{\varepsilon}}_e : \boldsymbol{\sigma}_c \, d\Omega, \quad (70)$$

where $\boldsymbol{\sigma}_e$ and $\boldsymbol{\sigma}_c$ are stress fields computed with the equilibrated and compatible models, respectively, and $\hat{\boldsymbol{\varepsilon}}_e$ is a virtual strain computed with the equilibrated model. We consider the mean value of (69) and (70):

$$\delta^{\hat{e}a} = \frac{1}{2}(\delta^{\hat{e}e} + \delta^{\hat{e}c}). \quad (71)$$

We introduce a norm of the error between two solutions:

$$\varepsilon = \sqrt{\int_{\Omega} (\boldsymbol{\varepsilon}_e - \boldsymbol{\varepsilon}_c) : (\boldsymbol{\sigma}_e - \boldsymbol{\sigma}_c) \, d\Omega}. \quad (72)$$

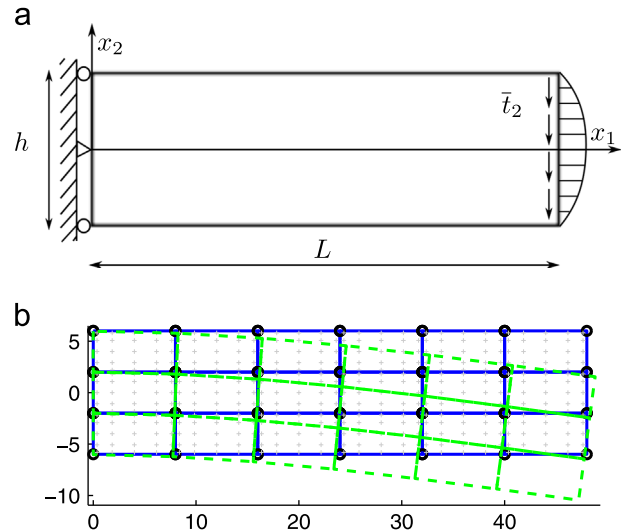


Fig. 8. Timoshenko beam problem. (a) Problem statement. (b) Particles distribution, integration cells and deformed shape.

Similarly, the same quantity between virtual solutions can be introduced as

$$\hat{\varepsilon} = \sqrt{\int_{\Omega} (\hat{\varepsilon}_e - \hat{\varepsilon}_c) : (\hat{\varepsilon}_e - \hat{\varepsilon}_c) d\Omega}. \quad (73)$$

It can be shown [7] that an exact value of the local displacement is bounded by

$$\delta^{\hat{\varepsilon}^a} \frac{1}{2} \hat{\varepsilon} \varepsilon < \delta < \delta^{\hat{\varepsilon}^a} + \frac{1}{2} \hat{\varepsilon} \varepsilon. \quad (74)$$

The essential part in obtaining this inequality is the usage of the equilibrated solution for the virtual strain. If one desires to define certain local stress, then the compatible solution for the prescribed virtual displacement should be considered.

5. Numerical examples

5.1. Timoshenko beam problem

Consider a beam subjected to a parabolic tip load as represented in Fig. 8a. The dimensions of the beam are $L=48$, $h=12$. The cross section of the beam is considered to have unit width. Material parameters are Young's modulus $E = 3 \times 10^7$ and Poisson's ratio $\nu = 0.3$.

The imposed tangential stress on the right edge is

$$\bar{t}_2 = \frac{P}{2I} \left(x_2^2 - \frac{h^2}{4} \right) \quad \text{where } I = \frac{h^3}{12}. \quad (75)$$

Hence, the total distributed load acting along the tip boundary renders

$$P = \int_{-h/2}^{h/2} \bar{t}_2 dx_2 = 1000. \quad (76)$$

The left edge of the beam is subjected to the kinematic boundary conditions defined by the exact solution for displacements:

$$u_1 = \frac{Py}{6EI} \left\{ (6L - 3x_1)x_1 + (2+\nu) \left(x_2^2 - \frac{h^2}{4} \right) \right\},$$

$$u_2 = \frac{P}{6EI} \left\{ 3\nu x_2^2 (L - x_1) + (4+5\nu) \frac{h^2 x_1}{4} + (3L - x_1) x_1^2 \right\}. \quad (77)$$

Implementation of both meshless models requires numerical integration and various strategies can be used to this end. The most straightforward way is to use conventional Gauss integration rules, which implies a mesh of so called integration cells to be built within the domain. In our examples we consider two types of cells, triangular (tri) and quadrilateral (quad), with different rules. An example of particles distribution and quad integration cells used for computation is presented in Fig. 8b, where the beam deformed shape is also shown.

One of the most important parameters required for the construction of the approximation functions is the MFLS basis degree, k . In this particular beam example $1 \leq k \leq 2$ was used for the compatible model and $2 \leq k \leq 3$ for the equilibrated one. Lagrange polynomials of the first degree for each particular case were chosen for boundary values approximations.

The system strain energy was the main global quantity analyzed during the computation. The exact solution of this problem is

$$U_{\text{exact}} = \frac{2LP^2(3h^2 + 5L^2 + 3h^2\nu)}{5h^3E} = 4.474667. \quad (78)$$

According to the results in Fig. 9a both models perfectly bound this value. Note that, for this problem, the exact solution is obtained if

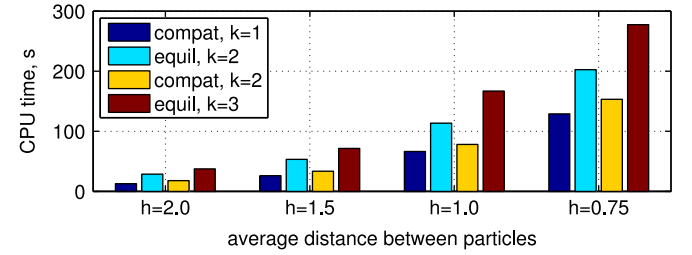


Fig. 10. Performance of compatible and equilibrated models.

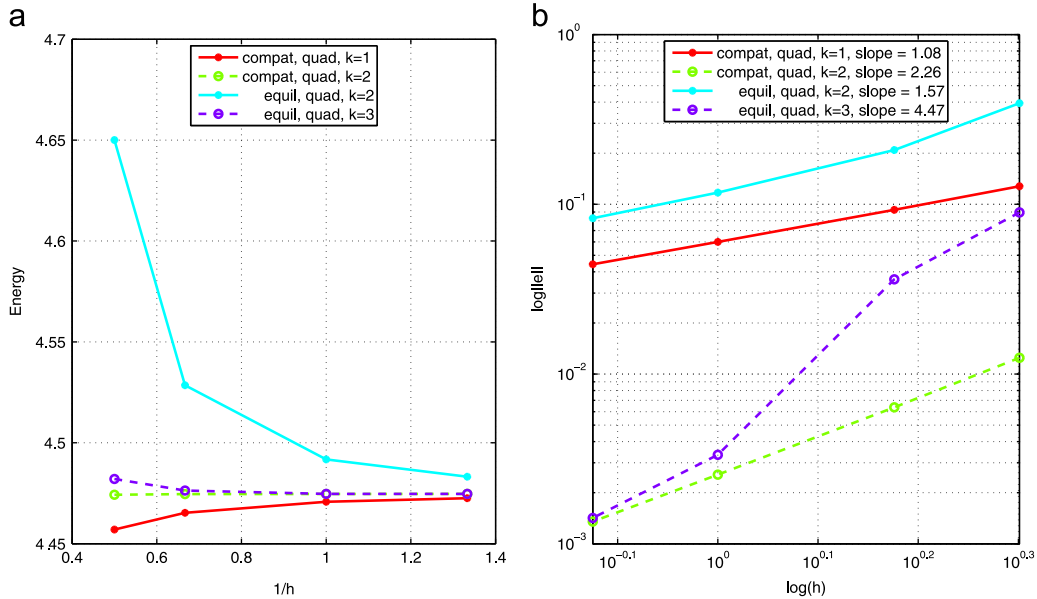


Fig. 9. Timoshenko beam. (a) Energy bounds and (b) convergence rates.

3rd and 4th degree polynomials are used in the MFLS basis for the compatible and the equilibrated models, respectively.

In order to estimate the convergence rates of the model, the following energy norm was used:

$$\|e\| = \sqrt{\int_{\Omega} (\mathbf{e}_{num} - \mathbf{e}_{exact}) : (\boldsymbol{\sigma}_{num} - \boldsymbol{\sigma}_{exact}) d\Omega}, \quad (79)$$

where \mathbf{e}_{num} and $\boldsymbol{\sigma}_{num}$ are the computed strain and stress fields respectively and \mathbf{e}_{exact} and $\boldsymbol{\sigma}_{exact}$ are the exact ones. The energy convergence rates (see Fig. 9b) for the compatible solution can be estimated [16] and are matched by our numerical solutions. However, for the equilibrated model the authors are not aware of such theoretical estimates.

The comparison of the performance of the two models reveals the main drawback of the equilibrated one: its high computational cost. Fig. 10 shows the corresponding CPU time, consumed in the construction of systems (25) and (35) and in obtaining their solution (the same domain discretization and integration points distribution are used for both models). As stated in Section 3.2, to get equal orders of approximation for the stress field, bases of higher degree need to be used for the equilibrated model, that, obviously, affects the CPU time. However, even if the performance for the same approximation bases is compared (see $k=2$ bars), the compatible model appears to be advantageous, mainly, due to two reasons:

- (i) higher derivatives of MFLS approximation functions are involved in the evaluation of the flexibility matrix (26a) (2nd order instead of 1st);
- (ii) larger size of matrix of constraints (36b), since for a particular domain, normally, the static boundary is larger than the kinematic one.

5.2. Infinite plate with a hole

Consider the problem of the infinite plate [17] with a central circular hole subjected to a unidirectional tensile load $p=1$ in x_1 direction. Plane stress conditions are assumed and Young's modulus is $E=1 \times 10^3$ and Poisson's ratio is $\nu=0.3$. Only the upper right quadrant of the plate is modeled due to symmetry, see Fig. 11a. To this end we apply on the right and top sides of the plate the stresses which correspond to exact solution of the problem:

$$\sigma_{11}(r, \theta) = 1 - \frac{a^2}{r^2} \left\{ \frac{3}{2} \cos 2\theta + \cos 4\theta \right\} + \frac{3a^4}{2r^4} \cos 4\theta,$$

$$\begin{aligned} \sigma_{22}(r, \theta) &= \frac{a^2}{r^2} \left\{ \frac{1}{2} \cos 2\theta - \cos 4\theta \right\} + \frac{3a^4}{2r^4} \cos 4\theta, \\ \sigma_{12}(r, \theta) &= \frac{a^2}{r^2} \left\{ \frac{1}{2} \sin 2\theta + \sin 4\theta \right\} + \frac{3a^4}{2r^4} \sin 4\theta, \end{aligned} \quad (80)$$

where (r, θ) are the polar coordinates and θ is measured from the positive x_1 axis counterclockwise.

Meshes of triangular and quadrilateral integration cells were built in the domain to obtain the governing system of equations. Bases with degrees ranging from 1 to 4 were assumed for both models and linear Lagrange polynomials were used for the boundary value approximations.

The exact value of the strain energy for this problem is $U_{exact} \simeq 0.0129754$, which is strictly bounded by the compatible and equilibrated solutions, see Fig. 12a. Both models demonstrate similar convergence rates comparing to the ones given by the previous example, as shown in Fig. 12b.

To show that the stresses in both cases coincide we provide Fig. 13, where the stresses σ_{11} on the left edge of the plate are shown. For this case 2nd and 3rd degree bases were used to build the compatible and the equilibrated solutions, respectively. Since the displacements in the x_1 direction are prevented on the left edge due to symmetry conditions we can also obtain the values of the Lagrange multipliers which correspond to tractions t_1 in the compatible model. These tractions nearly coincide with the values of stresses computed by both models, see Fig. 13. We also perform the similar comparison for the stress field σ_{11} on the entire domain, obtained by means of both models. Fig. 13 shows that they are very similar, but the compatible model gives slightly better results, since stresses are only first derivatives of domain variables, while in the equilibrated one they are defined by second derivatives.

The equilibrated model does not provide domain displacements, but the deformed shape of its boundary can be computed as we have either prescribed displacements on natural boundaries or values of the Lagrange multipliers on essential ones. The corresponding result is shown in Fig. 14a, which nearly coincides with Fig. 14b, where the displacements are computed with the compatible model. In the latter the sides of the integration cells were selected for the visualization.

We also compute the average displacement of the right edge of the plate, Γ_0 , that is marked in Fig. 11a. To this end we apply a virtual uniformly distributed tensile load along that boundary with

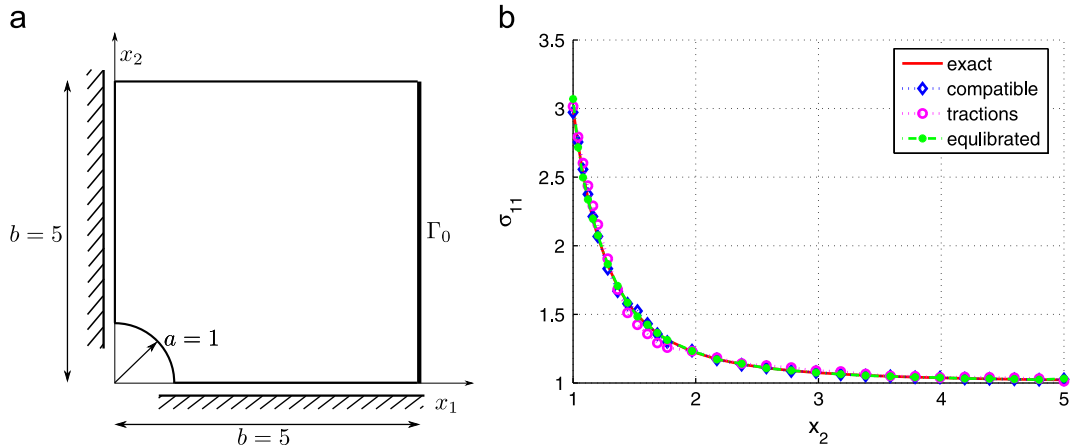


Fig. 11. Plate with a hole. (a) Problem statement. (b) Stress σ_{11} for $x_1 = 0$.

the magnitude of $1/\text{boundary length}$, i.e., $\frac{1}{5}$. The exact average displacement value is $\delta_{\text{exact}} = 5.340758 \times 10^{-3}$.

The quantity of interest computation process has an important implementation issue. In order to obtain its value two pairs of solutions are required, and each of them can be computed with different parameters, where the most important one is the degree of the basis k . To make the solution behavior more predictable we will set it for both models in such a way that the same degree of stress approximants is obtained. Therefore to get linear stress approximants the bases of degrees $k=2$ and $k=3$ should be chosen for the compatible model and for the equilibrated model, respectively. Consequently, to have quadratic stress approximants, the degrees of both bases must be increased by 1.

Tables 1 and 2 show results for the desirable quantity of interest, computed according to the methodology described in Section 4. We can observe that (i) computed bounds never intersect the exact value and (ii) the size of the interval reduces once the approximation is refined. The data, presented in Tables 1 and 2, is summarized in Fig. 15.

5.3. Perforated domain

Consider the perforated domain shown in Fig. 16a, where we compute an average displacement of the part Γ_0 of the domain's right edge following the same technique previously applied. The geometry of this example is no longer convex, resulting in the possibility of interaction of particles from the upper part of the domain with points of interest from the lower one. If the supports are small enough, then overlapping, when it occurs, does not affect the solution significantly, since the weight function tends to take small values when the point of interest is located at a distance $r > 0.5h_0$, where h_0 is a support size.

In this example only quadrilateral integration cells were used. Also we consider bases of degrees $2 \leq k \leq 3$ and $3 \leq k \leq 4$ for the compatible and equilibrated model, respectively. Linear Lagrange polynomials were used for the boundary approximants.

Tables 3 and 4 represent lower and upper bounds of the prescribed quantity of interest for different stress approximants respectively and Fig. 17 summarizes this data graphically. We can

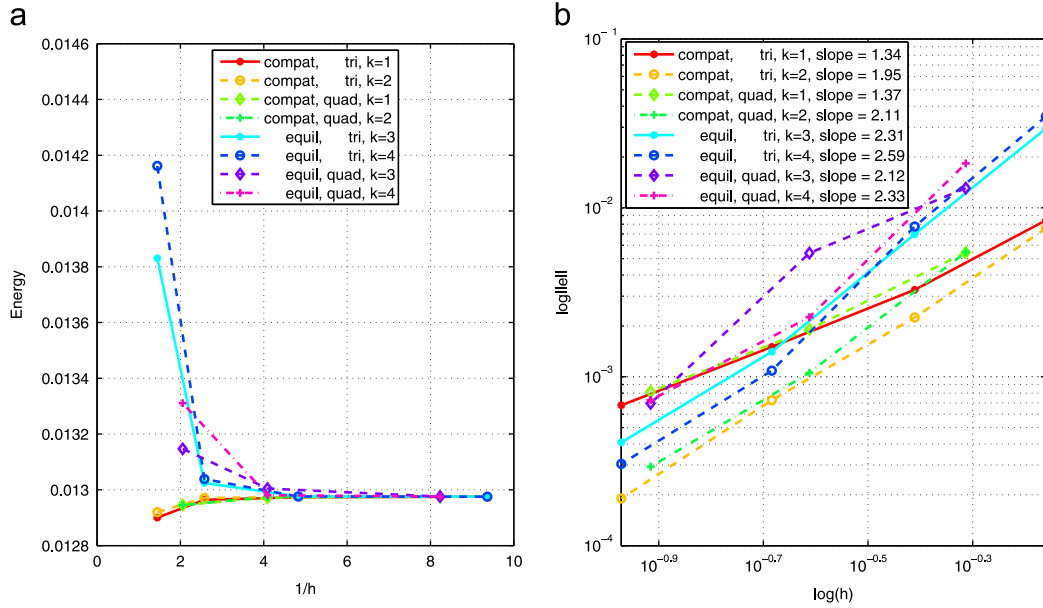


Fig. 12. Plate with a hole. (a) Energy bounds and (b) convergence rates.

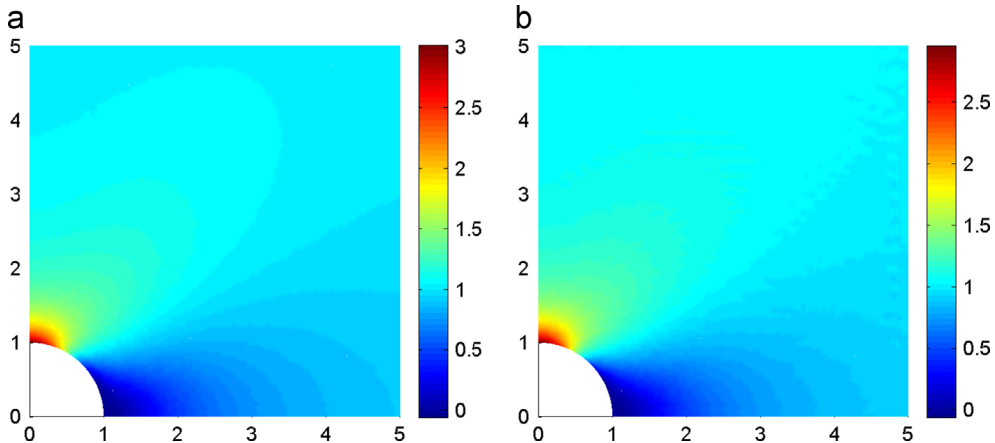


Fig. 13. Plate with a hole, stress σ_{11} . (a) The compatible model. (b) The equilibrated model.

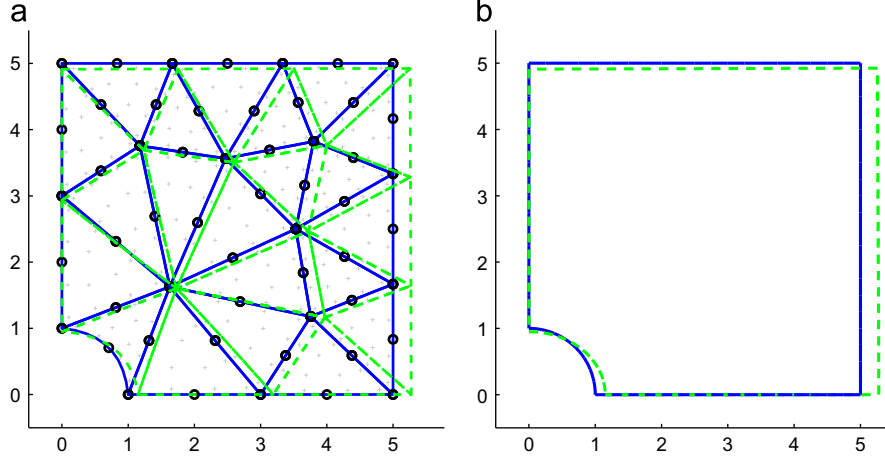


Fig. 14. Plate with a hole, deformed shape. (a) The compatible model. (b) The equilibrated model.

Table 1

Reduction of the size of the intervals bounding the exact displacement using linear stress approximants for the plate with a hole problem.

| h | δ_{lower}^{ea} | δ_{upper}^{ea} | $\hat{\epsilon}\epsilon$ |
|--------|---------------------------|---------------------------|---------------------------|
| 0.6919 | 5.310545×10^{-3} | 5.694433×10^{-3} | 3.838876×10^{-4} |
| 0.3883 | 5.336891×10^{-3} | 5.395936×10^{-3} | 5.904498×10^{-5} |
| 0.2070 | 5.340252×10^{-3} | 5.341371×10^{-3} | 1.119121×10^{-6} |
| 0.1070 | 5.340270×10^{-3} | 5.340790×10^{-3} | 5.201340×10^{-7} |

Table 2

Reduction of the size of the intervals bounding the exact displacement using quadratic stress approximants for the plate with a hole problem.

| h | δ_{lower}^{ea} | δ_{upper}^{ea} | $\hat{\epsilon}\epsilon$ |
|--------|---------------------------|---------------------------|---------------------------|
| 0.6919 | 5.313955×10^{-3} | 5.724771×10^{-3} | 4.108154×10^{-4} |
| 0.3883 | 5.337428×10^{-3} | 5.419258×10^{-3} | 8.183068×10^{-5} |
| 0.2070 | 5.340393×10^{-3} | 5.341424×10^{-3} | 1.031118×10^{-6} |
| 0.1070 | 5.340508×10^{-3} | 5.340782×10^{-3} | 2.740336×10^{-7} |

observe that the bounds tend to demonstrate a behavior similar to what was obtained for the previous example.

Analyzing the slope of the interval size reduction in both Figs. 15b and 17b, one can observe that the degree of the basis used to build the approximation does not improve significantly the results, unlike what would be expected, based on the experience with similar plots for FEM models. But such direct comparison between the FEM p refinement and the MFLS approximants basis increment is misleading. The FEM p refinement assumes that the number of elements is kept constant but the number of degrees of freedom within each element increases according to the higher order of the element, resulting in a larger number of degrees of freedom. On the other hand, when the degree of the basis used to build MFLS approximation functions is increased, the number of degrees of freedom is kept unaltered. Therefore the change of the basis for the meshless approximants cannot be directly compared with the conventional p refinement in the FEM.

6. Conclusions

We successfully exploited the possibility of using a dual analysis approach by means of meshless methods, with multiple fixed least squares approximations used as the core for the discretization process. Both equilibrated and compatible models were presented: the former was given in more detail, while the latter was shortly highlighted referring to previous well known works. The dual approach was successfully used for bounds computation of both global and local quantities in several numerical examples.

The main implementation issues, generally associated to the equilibrated model, were outlined. In order to avoid the identified difficulties, the following requirements are mandatory for a robust implementation:

- (i) To prevent ill conditioning of the moment matrix, \mathbf{A} , used to set up the MFLS approximation functions, the domain discretization demands, even for simple cases, a larger number of particles in comparison with the compatible model. This requirement arises due to the high degree of the basis used to build the equilibrated solution. We remark that, even though for the same accuracy the equilibrated model requires more particles, the number of degrees of freedom is comparable with the compatible one, since it has one DOF per particle instead of two.
- (ii) The natural ill conditioning of the flexibility matrix should be explicitly considered.
- (iii) Some dependencies may arise on the essential boundary. They should be eliminated, for instance, by means of the introduction of extra Lagrange multipliers. In fact, this problem also appears in the compatible solution, but since the static boundary is usually larger than the kinematic one, the equilibrated model is much more prone to this kind of problems. Care should also be taken to ensure the admissibility of the non homogeneous boundary conditions.

An attempt to impose the EBC locally was performed. This idea leads to the usage of the boundary traces of the domain approximants as the Lagrange multipliers approximation basis for the EBC enforcement. Once the suggested approach was implemented, problems related to numerical stability of the obtained system of equations arose. The inf sup test confirmed the observed problem. It induces one to use totally different function spaces for the

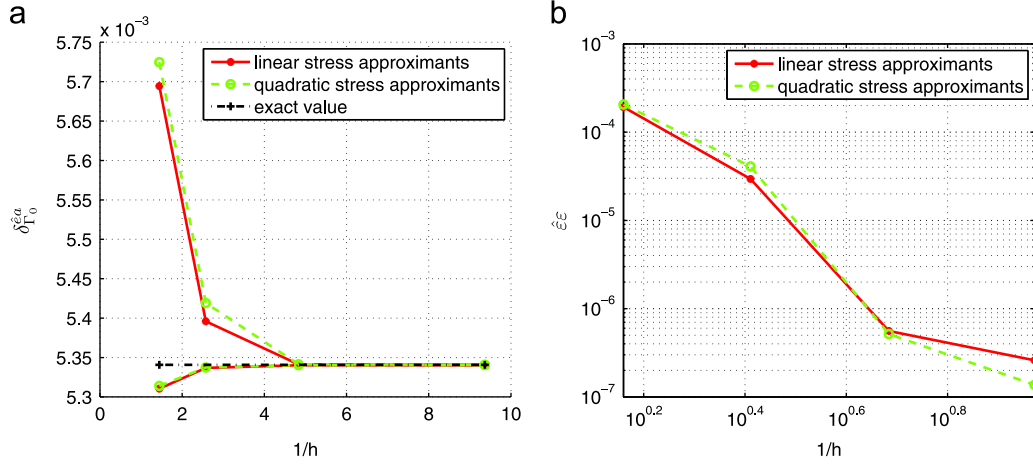


Fig. 15. Plate with a hole. Right edge displacement upper and lower bounds. (a) Displacement value. (b) Interval size.

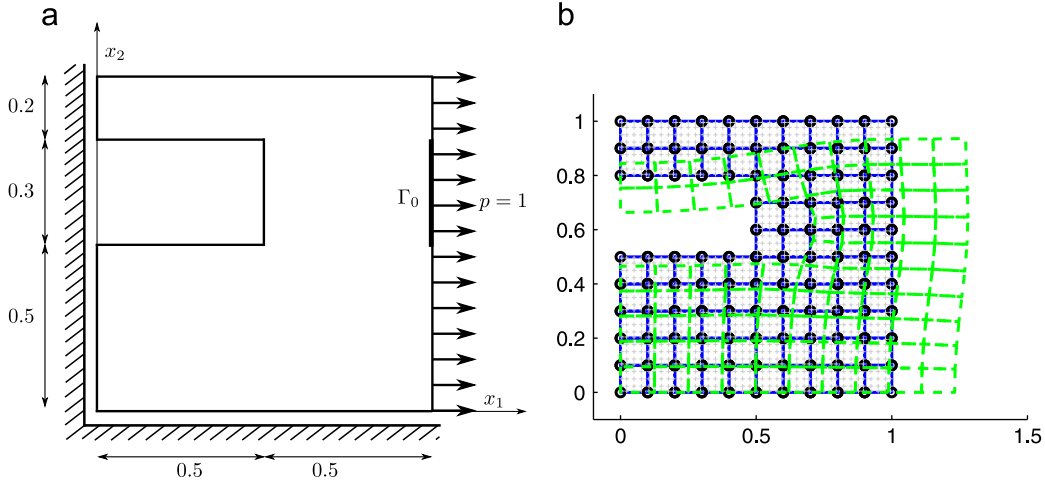


Fig. 16. Perforated domain problem. (a) Problem statement. (b) Particles distribution, integration cells and deformed shape.

Table 3

Reduction of the size of the intervals bounding the exact displacement using linear stress approximants for the perforated domain problem.

| h | $\delta_{lower}^{\epsilon a}$ | $\delta_{upper}^{\epsilon a}$ | $\hat{\epsilon}\epsilon$ |
|-------|-------------------------------|-------------------------------|---------------------------|
| 0.100 | 1.344377376 | 2.220901954 | 8.765246×10^{-1} |
| 0.050 | 1.398469053 | 1.448217192 | 4.974814×10^{-2} |
| 0.033 | 1.405630965 | 1.429631618 | 2.400066×10^{-2} |
| 0.025 | 1.407592593 | 1.422059353 | 1.446676×10^{-2} |

Table 4

Reduction of the size of the intervals bounding the exact displacement using quadratic stress approximants for the perforated domain problem.

| h | $\delta_{lower}^{\epsilon a}$ | $\delta_{upper}^{\epsilon a}$ | $\hat{\epsilon}\epsilon$ |
|-------|-------------------------------|-------------------------------|---------------------------|
| 0.100 | 1.361946142 | 1.802515749 | 4.405696×10^{-1} |
| 0.050 | 1.397846169 | 1.474901772 | 7.705560×10^{-2} |
| 0.033 | 1.404219921 | 1.436820552 | 3.260064×10^{-2} |
| 0.025 | 1.406491014 | 1.424092154 | 1.760114×10^{-2} |

domain and for the boundary approximations in the case of the meshless methods, making strict EBC enforcement impossible for the proposed techniques. Nevertheless, the resulting lack of equilibrium which may appear on the essential boundary is negligible and can be disregarded.

Once these limitations are known, dual analysis can be performed efficiently. The presented numerical examples demonstrate the robustness of the described approach:

- (i) the exact value of the strain energy is perfectly bounded by the compatible and the equilibrated models;
- (ii) the bounds of the error of local quantities are also strict;
- (iii) the energy error reduces once the domain particles distribution is refined, demonstrating adequate rates of convergence;
- (iv) the results do not depend significantly on the integration cells type, thus confirming the precision of the computed integrals;
- (v) the graphical representation of the deformed shapes and stress fields of both models reveals a close matching.

The stress model cannot replace the displacement one due to the reasons mentioned above but it can serve as a natural verification method for global (strain energy) and local (stresses

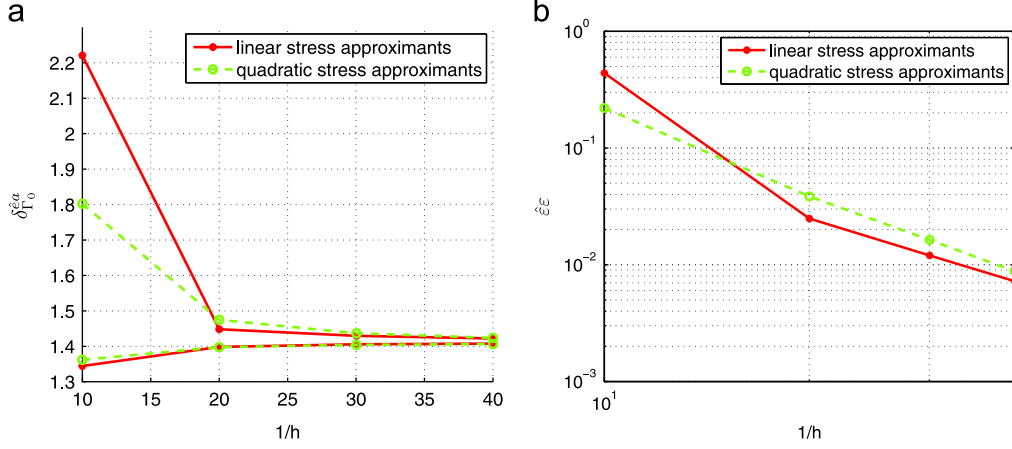


Fig. 17. Perforated domain. Right edge displacement upper and lower bounds. (a) Displacement value. (b) Interval size.

and displacements) quantities of interest, bringing confidence to the obtained solution.

Acknowledgments

The first author thanks the support of the European Union as an Early Stage Researcher in the Marie Curie Initial Training Network (FP7 PEOPLE ITN 2008) ATCoMe Advanced Techniques in Computational Mechanics, Project no. 238548.

The work of the team at IST is also part of the research activity carried out at ICIST, Instituto de Engenharia de Estruturas, Território e Construção, and has been partially financed by FCT (Fundação para a Ciência e Tecnologia) in the framework of Project ICIST, U0076.

Appendix A. The MATLAB code skeleton

To solve an arbitrary 2D plane elasticity problem by either of two models, it is necessary to build an input file, defining:

- (i) domain configuration as a closed boundary with internal particles distribution, `particles`,
- (ii) underlying integration grid, `gaussPoints` not a mandatory operation, since the grid can be automatically constructed by means of, e.g., a Delaunay triangulation, once the particles distribution is defined,
- (iii) static and kinematic boundary conditions that are later assigned as natural and essential ones for the compatible model and *vice versa* for the equilibrated one,

- (iv) body forces,
- (v) MFLS approximation function parameters, i.e., basis degree k , support size scale factor d , a type of weight function `wfunction` and a type of particles support (elliptical or rectangular),
- (vi) material parameters.

Thus, the implementation of both models is done within the same program, where certain minor aspects are programmed independently. The general flowchart of the code looks as:

```
% 1. Input file reading
readdata(filename);
% 2. Some basic initializing
initialize(..);
% 3. Generation of integration points
gaussPoints = integration(..);
% 4. Evaluation of supports sizes
supports = covering(..);
% 5. Stiffness/flexibility matrix
[mtrK, vecF] = stiffness(..);
% 6. Natural and essential BCs
[mtrG, vecQ, mtrC, vecF] = boundaries(..);
% 7. Assign Airy stress function initial values
% see Section 3.4 – for equilibrated model
[mtrR, vecS] = airy(..);
% 8. Solution
vecX = solver(..);
```

Some key components of the program are presented below as a pseudo code.

Assignment of supports sizes is done according to methodology, described in [Section 3.2](#):

```
% particles — array of particles
% 1,2 — x,y coordinates
% gaussPoints — array of Gauss pts
% 1,2 — x,y coordinates; 3 — weight, 4 — jacobian
function supports = covering(particles, ...
    gaussPoints, basis, suptype, dfactor)

nnodes = size(particles,2);
ncovered = 0.5*(basis+1)*(basis+2);
supports = zeros(2,nnodes);
for i=1:size(gaussPoints,2)
    numPosGauss = [0; gaussPoints(1:2,i)];
    nDiff = particles(1:3,:) - ...
        numPosGauss*ones(1,nnodes);
    nDist = zeros(2,nnodes);
    nDist(1,:) = nDiff(1,:);
    nDist(2,:) = sqrt(nDiff(2,:).^2+nDiff(3,:).^2);
    nSort = sortrows(nDist',2)';
    for j=1:ncovered
        if strcmp(suptype, 'elliptical')
            if supports(1,nSort(1,j)) < nSort(2,j)
                supports(1,nSort(1,j)) = nSort(2,j);
                supports(2,nSort(1,j)) = nSort(2,j);
            end;
        else
            if supports(1,nSort(1,j)) < ...
                abs(nDiff(2,nSort(1,j)))
                supports(1,nSort(1,j)) = ...
                    abs(nDiff(2,nSort(1,j)));
            end;
            if supports(2,nSort(1,j)) < ...
                abs(nDiff(3,nSort(1,j)))
                supports(2,nSort(1,j)) = ...
                    abs(nDiff(3,nSort(1,j)));
            end;
        end;
    end;
end;
supports = dfactor*supports;
```

The following function constructs stiffness/flexibility matrix:

```
function [mtrK, vecF] = stiffness(gaussPoints, ...
    particles, supports, material, loads, ...
    wfunction, basis, domainDofsNum)

mtrK = zeros(domainDofsNum,domainDofsNum);
vecF = zeros(domainDofsNum,1);

% Loop over gauss points
for gpoint=gaussPoints
    position = gpoint(1:2);
    weight = gpoint(3);
    jacobian = gpoint(4);

    % Evaluate all non zero nodal functions and ...
    % derivatives at gauss point
    [Phi, idsDmn] = MFLSapproximation(position, ...
        particles, supports, wfunction, basis);
    % Evaluate domain stiffness/flexibility and ...
    % force vector for gauss point
    [gpK, gpF] = gpointStiffness(material, ...
        loads, Phi);

    mtrK(idsDmn,idsDmn) = mtrK(idsDmn,idsDmn) + ...
        weight*jacobian*gpK;
    vecF(idsDmn) = vecF(idsDmn) + ...
        weight*jacobian*gpF;
end;
```

where function `gpointStiffness()` implements either [\(26a\)](#) or

[\(36a\)](#), depending on the model. Function `MFLSapproximation()` is a standard implementation of MFLS/MLS procedure [\[4,22\]](#) that for a given point of interest constructs an approximation function and its derivatives up to 1st order for the compatible model and up to 2nd order for the equilibrated one.

A single function is responsible for imposition of both natural and essential boundary conditions:

```
function [mtrG, vecQ, mtrC, vecF] = ...
    boundaries(boundaries, corners, cells, ...
        particles, supports, basis, wfunction)

mtrG = []; % Matrix of constraints
vecQ = []; % Vector of imposed EBCs
mtrC = []; % Matrix to release corner linear ...
    dependencies
vecF = zeros(domainDofsNum,1); % Vector of ...
    imposed NBCs

for j=1:length(boundaries)
    boundary = boundaries(j);

    % Find cells, that touch the boundary
    boundary = findBoundaryCells(cells, boundary);
    % Split boundary into integration cells
    vcsi = splitBoundary(boundary);

    if boundary.type == 'essential'
        % If EBCs are imposed in x and y directions
        imposed = boundary.onx+boundary.ony;

        Q = zeros(imposed*nnum,1);
        G = zeros(domainDofsNum,imposed*nnum);
    end;

    for isub=1:boundary.divisions

        % Generate gauss points
        gaussld = gaussld(boundary, isub, vcsi);

        for gpoint = gaussld
            gposld = gpoint(1);
            weight = gpoint(2);
            jacobld = gpoint(3);
            csi = gpoint(4);

            % Map point from parametric space to ...
            % physical
            [xq, yq, jac, nx, ny] = ...
                mapToBoundary(cells, boundary, ...
                    gposld);
            gpos2d = [xq; yq];
            jac = jac*jacobld;

            % Matrix of boundary normal
            Normal = [nx 0 ny; 0 ny nx];

            % Evaluate contributing DOMAIN ...
            % approximation function values ...
            % for the Gauss point
            [Phi, idsDmn] = ...
                MFLSapproximation(gpos2d, ...
                    particles, supports, wfunction, ...
                    basis);
```

```

if boundary.type == 'essential'
    % Evaluate contributing BOUNDARY ...
    approximation function ...
    values for the Gauss point
    % can be Lagrange polynomials or ...
    MFLS traces
    [Xi, idsBnd] = ...
        boundaryApprox(particles, ...
            supports, csi, boundary, ...
            imposed);

    [gpG, gpQ] = ...
        gpointConstraints(boundary, ...
            imposed, Xi, Phi, Normal);

    Q(idsBnd) = Q(idsBnd) - ...
        weight*jac*gpQ;
    G(idsDmn,idsBnd) = ...
        G(idsDmn,idsBnd) - ...
        weight*jac*gpG;
else
    gpF = gpointNatural(boundary, ...
        Phi, Normal);

    vecF(idsDmn) = vecF(idsDmn) + ...
        weight*jac*gpF;
end;
end;

if boundary.type == 'essential'
    mtrG = [mtrG, mtrG];
    vecQ = [vecQ, vecQ];
end;

% Extra Lagrange multipliers to release ...
dependencies at corners – see section 3.5
dofsNum = 0;
for j=1:length(corners)
    dofsNum = dofsNum + corners(j).onx + ...
        corners(j).ony;
end;
mtrC = zeros(sum(size(mtrG)),dofsNum);

for j=1:length(corners)
    mtrC = ...
        connectCorners(boundaries(corners(j).b1), ...
            boundaries(corners(j).b2));
end;

```

If the equilibrated model is being solved, as explained in (3.4), Airy stress function is constrained to be zero at three predefined points:

```

function [mtrR, vecS] = airy(anodes)

nvals = 3;
mtrR = zeros(nvals,size(mtrK,2));
vecS = zeros(nvals,1);
for i=1:length(data.airy)
    mtrR(i,anodes(i)) = -1;
end;

```

The construction of resulting systems of Eqs. (25) and (35) and their solution is done in a trivial way:

```

function vecX = solver(mtrK, mtrG, mtrC, mtrR, ...
    vecF, vecQ, vecC, vecS)

mtrP = [mtrR; zeros(size(mtrG,2)+size(mtrC,2), ...
    size(mtrR,2))];

B = [vecF; vecQ; vecC; vecS];

% Stiffness/flexibility and Lagrange multipliers
A = [mtrK mtrG;
    mtrG' zeros(size(mtrG,2))];
% Resolve possible corner linear dependencies
A = [ A mtrC;
    mtrC' zeros(size(mtrC,2))];
% Fix three Airy stress function values
A = [ A mtrP;
    mtrP' zeros(size(mtrP,2))];

vecX = A\B;

```

Once the solution of two models is available, an implementation of the local quantities of interest evaluation strategy, given in Section 4, becomes straightforward:

```

function [quantity, interval] = qoi()
% What is our QOI – displacement or stress
compute = 'displacement';

% Input files for the problem
file.compat_pri = 'example-compatible_pri.xml';
file.equil_pri = 'example-equilibrated_pri.xml';
file.compat_sec = 'example-compatible_sec.xml';
file.equil_sec = 'example-equilibrated_sec.xml';

% Solve 2 models for given and unitary loads
compat_pri = compatible(file.compat_pri);
compat_sec = compatible(file.compat_sec);
equil_pri = equilibrated(file.equil_pri);
equil_sec = equilibrated(file.equil_sec);

% Now compute quantity and error
quantity = 0; error_pri = 0; error_sec = 0;
for gpoint=compat_pri.gaussPoints
    gpos = gpoint(1:2);
    weight = gpoint(3);
    jac = gpoint(4);

    [Phi1, ids1] = MFLSapproximation(gpos, ...
        compat_pri.particles, ...
        compat_pri.supports, ...
        compat_pri.wfunction, compat_pri.basis);
    [Phi2, ids2] = MFLSapproximation(gpos, ...
        equil_pri.particles, equil_pri.supports, ...
        equil_pri.wfunction, equil_pri.basis);
    [Phi3, ids3] = MFLSapproximation(gpos, ...
        compat_sec.particles, ...
        compat_sec.supports, ...
        compat_sec.wfunction, compat_sec.basis);

```

```

[Phi4, ids4] = MFLSapproximation(gpos, ...
    equil_sec.particles, equil_sec.supports, ...
    equil_sec.wfunction, equil_sec.basis);
diff_stress_pri = ...
    stressEquil(equil_pri, Phi2, ids2) - ...
    stressCompat(compat_pri, Phi1, ids1);
diff_stress_sec = ...
    stressEquil(equil_sec, Phi4, ids4) - ...
    stressCompat(compat_sec, Phi3, ids3);
error_pri = error_pri + weight*jac * ...
    diff_stress_pri' * ...
    (equil_pri.invD*diff_stress_pri);
error_sec = error_sec + weight*jac * ...
    diff_stress_sec' * ...
    (equil_pri.invD*diff_stress_sec);

if compute == 'displacement'
    quantity = quantity + weight*jac * ...
        stressEquil(equil_sec, Phi4, ids4)' * ...
        (strainCompat(compat_pri, Phi1, ids1) ...
        + strainEquil(equil_pri, Phi2, ids2));
else
    quantity = quantity + weight*jac * ...
        stressEquil(compat_sec, Phi3, ids3)' * ...
        (strainCompat(compat_pri, Phi1, ids1) ...
        + strainEquil(equil_pri, Phi2, ids2));
end;
end

quantity = quantity/2;
interval = sqrt(error_pri)*sqrt(error_sec)/2;

```

In the listing above the evaluation of the corresponding QoI for the problem with the homogeneous essential boundary conditions is presented; however, computations with non homogeneous ones do not face any significant difficulties and just involve slightly more manipulations with boundary integrals, which are similar to implementation of function boundaries(). Functions strainCompat(), strainEquil(), stressCompat() and stressEquil() evaluate strains and stresses for the compatible and the equilibrated models respectively.

References

- [1] Babuška I, Banerjee U, Osborn JE. Meshless and generalized finite element methods: a survey of some major results In: Griebel Michael, Schweitzer Marc Alexander, editors. Meshfree methods for partial differential equations. Lecture notes in computational science and engineering, vol. 26. Berlin, Heidelberg: Springer; 2002. p. 1–20.
- [2] Bathe K-J. Finite element procedures. Prentice Hall; 1996.
- [3] Bathe K-J, Brezzi F. Studies of finite element procedures. The inf-sup condition, equivalent forms and applications. In: Reliability of methods for engineering analysis. UK: University College of Swansea; 1986.
- [4] Belytschko T, Lu YY, Gu L. Element-free Galerkin methods. International Journal for Numerical Methods in Engineering 1994;37(2):229–56.
- [5] Belytschko T, Organ D, Krongauz Y. A coupled finite element–element-free Galerkin method. Computational Mechanics 1995;17(3):186–95.
- [6] De S, Bathe K-J. Displacement/pressure mixed interpolation in the method of finite spheres. International Journal for Numerical Methods in Engineering 2001;51(3):275–92.
- [7] Moitinho de Almeida JP, Almeida Pereira OJB. Upper bounds of the error in local quantities using equilibrated and compatible finite element solutions for linear elastic problems. Computer Methods in Applied Mechanics and Engineering 2006;195(4–6):279–96.
- [8] Fraeijs de Veubeke B. Upper and lower bounds in matrix structural analysis. In: AGARDograph 72: matrix methods of structural analysis. Pergamon Press: London; 1963. p. 165–201.
- [9] Fraeijs de Veubeke B. Displacement and equilibrium models in the finite element method. In: Zienkiewicz OC, Holister GS, editors. Stress analysis. John Wiley & Sons; 1965. p. 145–97.
- [10] Debongnie JF, Beckers P, Zhong HG. Dual analysis with general boundary conditions. Applied Mechanics and Engineering 1995;122:183–92.
- [11] Duflot M. Application des méthodes sans maillage en mécanique de la rupture. PhD thesis. Université de Liège; 2004.
- [12] Duflot M, Nguyen-Dang H. Dual analysis by a meshless method. Communications in Numerical Methods in Engineering 2002;18(9):621–31.
- [13] Fernández-Méndez S, Huerta A. Imposing essential boundary conditions in mesh-free methods. Computer Methods in Applied Mechanics and Engineering 2004;193(12–14):1257–75.
- [14] Griebel M, Schweitzer MA. A particle-partition of unity method—Part V: boundary conditions. In: Hildebrandt S, Karcher H, editors. Geometric analysis and nonlinear partial differential equations. Springer; 2002. p. 517–40.
- [15] Ladevèze P, Maunder EAW. A general method for recovering equilibrating element tractions. Computer Methods in Applied Mechanics and Engineering 1996;137(2):111–51.
- [16] Li S, Liu WK. Meshfree particle methods. Springer; 2004.
- [17] Liu GR. Mesh free methods: moving beyond the finite element method. CRC Press; 2003.
- [18] Liu GR. Smoothed finite element methods. Taylor & Francis; 2010.
- [19] Moitinho de Almeida JP, Maunder EAW. Recovery of equilibrium on star patches using a partition of unity technique. International Journal for Numerical Methods in Engineering 2009;79(12):1493–516.
- [20] Nayroles B, Touzot G, Villon P. Generalizing the finite element method: diffuse approximation and diffuse elements. Computational Mechanics 1992;10(5):307–18.
- [21] Neto EAS, Perić PD, Neto EAS, Owen PDRJ, Owen DRJ. Computational methods for plasticity: theory and applications. John Wiley & Sons; 2008.
- [22] Nguyen VP, Rabczuk T, Bordas S, Duflot M. Meshless methods: a review and computer implementation aspects. Mathematics and Computers in Simulation 2008;79(3):763–813.
- [23] Oñate E, Idelsohn S, Zienkiewicz OC, Taylor RL, Sacco C. A stabilized finite point method for analysis of fluid mechanics problems. Computer Methods in Applied Mechanics and Engineering 1996;139(1–4):315–46.
- [24] Tiago C. Meshless methods: extending the linear formulation and its generalization to geometrically exact structural analysis. PhD thesis. Instituto Superior Técnico, Technical University of Lisbon; 2007.
- [25] Tiago C, Pimenta PM. An EFG method for the nonlinear analysis of plates undergoing arbitrarily large deformations. Engineering Analysis with Boundary Elements 2008;32(6):494–511.
- [26] Timoshenko SP, Goodier SN. Theory of elasticity. New York: McGraw-Hill; 1987.
- [27] Zhu T, Atluri SN. A modified collocation method and a penalty formulation for enforcing the essential boundary conditions in the element free Galerkin method. Computational Mechanics 1998;21(3):211–22.